The flapping flight of birds

Thielicke, William

Chapter II

Digital Particle Image Velocimetry

This chapter is based on a manuscript submitted to the Journal of Open Research Software:
W. Thielicke and E.J. Stamhuis
PIVlab – Towards user-friendly, affordable and accurate digital particle image velocimetry in MATLAB

RATIONALE

Digital Particle Image Velocimetry (DPIV) is a common technique for non-intrusive, quantitative and qualitative flow visualization that will be introduced in short in this chapter. The chapter is intended to:

a provide a description of the material and methods used in this thesis
b give an evaluation of the accuracy of these methods
c supply background information for future studies and identify limitations

Description of the material and methods used in this study

One of the most important methods in this thesis is DPIV. At the beginning of the studies summarized in this thesis, an adequate DPIV tool capable of automatic analysis of huge amounts of data was not available in the lab. For the compilation of time-resolved, three-dimensional datasets, automatic analyses are essential. A high degree of automation can be achieved by tailoring a custom DPIV tool according to the specific needs of the project. Existing DPIV tools are mostly 'black boxes', which make it difficult to understand the details of the analysis, to judge the quality of the results and to automate the analyses. Developing a custom tool enables the automation of analyses, as well as the understanding of the principles and the challenges of DPIV in detail. Working 'behind the scenes' does also allow tweaking the experimental setup and the analysis methods for an optimal result. Literature was scanned, and the most promising and suitable principles were incorporated in the DPIV tool `PIVlab`[1] which was developed during the research presented in this thesis. The descriptions of the methods in the succeeding research chapters are kept as short as possible, following the conventional form. However, `PIVlab` was developed virtually from scratch, and the choice of adequate methods is crucial for getting reliable and repeatable results.

Evaluation of the accuracy of these methods

A number of research articles deals with the implementation and optimization of the DPIV technique (e. g. Keane & Adrian, 1990; Adrian, 1991; Willert & Gharib, 1991; Buchhave, 1992; Stamhuis & Videler, 1995; Willert, 1996; Grant, 1997; Raffel et al., 2007). The amount of literature dealing with this subject implies that already small differences in the methods may alter the results. DPIV is sensitive to computational details such as image

1 available at http://PIVlab.blogspot.com

pre-conditioning, different concepts of image correlation, subpixel peak estimators or data interpolation. Therefore, previous results about DPIV accuracy do not necessarily apply to the DPIV tool presented in this thesis. This chapter contains a section on the accuracy of PIVlab that analyses the influence of a selection of parameters. These accuracy tests are essential to confirm the quality of the measurements and results.

Background information for future studies and identification of limitations

The DPIV tool developed for this project is open source, a rapidly increasing number of studies uses PIVlab for analysing particle or pattern movements[2]. All studies that use PIVlab – including the thesis presented here – need an in-depth analysis of the underlying principles and an evaluation of the measurement accuracy. Factors that potentially limit the applicability also need to be determined. The latter is especially important, as PIVlab is often used for analyses of particle movements in a very different context than it was originally intended for (e. g. flow visualizations within animal cells, deformation of sand and gravel). This chapter may therefore serve as a reference for all studies conducted with PIVlab.

The order of content of this chapter follows the sequence of a typical DPIV experiment: First, the physical setup is illustrated, followed by an illustration of the analysis of DPIV image data. Finally the accuracy of DPIV measurements is evaluated, and further post-processing techniques are presented and evaluated.

2 e. g. Leong et al. (2011); Booth-Gauthier et al. (2012); Jiang & Towhata (2012); Piro et al. (2012); Roy et al. (2012); Ryerson & Schwenk (2012); Sanchez et al. (2012); Todaro et al. (2012); Cambau et al. (2013); Chen et al. (2013); Datta et al. (2013); Eriksson et al. (2013); Miller et al. (2013); Panciroli & Porfiri (2013); Senatore et al. (2013); Simmons et al. (2013); Sun & Hsiao (2013); Taddeucci et al. (2013); Wang et al. (2013); Zhou et al. (2013); Bloch et al. (2014); Catarino et al. (2014); Di Domenica et al. (2014); Hartmann et al. (2014); Heller et al. (2014); Jalalisendi et al. (2014); Maxwell et al. (2014); Melnikov et al. (2014); Schlüßler et al. (2014); Shi et al. (2014); e Silva et al. (2014); Sipos et al. (2014); Trojanowski et al. (2014); Wu et al. (2014); Cabrera et al. (2015)

IMAGE ACQUISITION

In DPIV, the flow is visualized by illuminating a thin sheet of fluid containing reflective and neutrally buoyant tracer particles. A digital image sensor is positioned parallel to the illuminated sheet, capturing the movement of the particles (see Figure 2.1). In most DPIV analyses, two images (A and B) of the illuminated plane are captured at t_o and $t_o + \Delta t$. Velocities in the sheet can hence be derived from Δt and the distance that the particles travelled from image A to B (particle displacement). In DPIV, the particle displacement is calculated for groups of particles by evaluating the cross-correlation of many small sub images (interrogation areas). The correlation yields the most probable displacement for a group of particles travelling on a straight line between image A and image B (Raffel et al., 2007).

PARTICLES

Fluids are in most cases homogeneous, and therefore velocities are hard to measure directly via optical means. DPIV hence relies on small reflective particles that are added to the fluid. These particles have to be of similar density as the fluid, in order to have the same inertia and buoyancy as the fluid under test. When doing measurements in water, finding suitable particles is often not problematic, whereas measurements in air require more attention because of the low density of this fluid (Stamhuis, 2006; Raffel et al., 2007). In a continuously accelerating element of fluid, the velocity lag U_s between

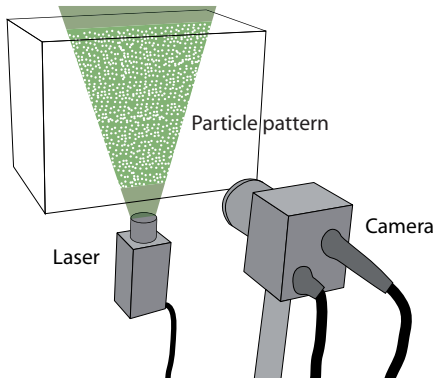

Fig. 2.1: Principle of DPIV: A laser sheet illuminates the particles contained in the fluid. A high-speed camera records the displacement of the particle pattern.

particle velocity U_p and fluid velocity U can, according to Stokes drag law, be expressed as (Raffel et al., 2007):

$$U_s = U_p - U = d_p^2 \frac{\rho_p - \rho}{18\mu} a \tag{2.1}$$

where d_p = particle diameter; ρ_p = particle density; ρ = fluid density; μ = fluid dynamic viscosity; a = acceleration

Minimizing the difference between fluid and particle velocity can therefore be achieved by reducing particle size, reducing the difference in density, increasing dynamic viscosity of the fluid or minimizing acceleration. The latter two factors can hardly be influenced without substantially changing the experimental conditions. Matching particle and fluid density is number one choice in liquids but often not possible for gaseous fluids (Raffel et al., 2007). Minimizing particle size is the most convenient solution, but decreases the amount of reflected light and therefore the signal to noise ratio of the captured images (Melling, 1997). Finding a suitable particle size is hence a tradeoff between a number of demands. The measurements in this thesis are conducted in water at 20° C using polyamide particles with a diameter of about 57 µm and a density of 1140 kg/m³, which is very close to the density of water, hence U_s is expected to be negligible.

ILLUMINATION AND CAMERA

Lasers are typically preferred for illumination in DPIV studies (Adrian, 1991). Most lasers deliver very high power, parallel light, which can easily be transformed to a thin sheet using one or more cylindrical lenses (Stamhuis, 2006). A wide variety of laser types that use different laser materials and energy pump sources is available on the market. Due to a very compact size and the high efficiency, diode pumped solid state lasers (DPSS) got more and more attention in the DPIV community (Raffel et al., 2007). Today's most important DPSS lasers are Neodym-Yttrium-Aluminium-Garnet (Nd:YAG) lasers. After frequency doubling, the beam that finally leaves the laser head has a wavelength of 532 nm. DPSS lasers can be operated in pulsed mode or in constant wave (CW) mode. Pulsed lasers achieve very high power peaks during very short pulses and make it possible to reduce the effective exposure time of the camera. Usually, CW lasers require a longer exposure to meet the light intensity requirements, which can lead to excessive motion blur. This can become problematic for the quality of the cross-correlation (Nobach & Honkanen, 2005). The advantage of using CW lasers is a less complicated setup, as there is no need to synchronize the high-speed camera with the laser pulses. The DPIV experiments in this thesis are conducted using a 5W CW Nd:YAG DPSS laser (Snoc electronics co., LTD, Guangdong, China) emitting light with a wavelength of 532 nm. The laser sheet is conditioned using two spherical lenses for adjusting the beam diameter, and a cylindrical lens to generate the sheet.

In high speed / time-resolved DPIV systems, cameras with CMOS imaging sensors became state of the art, due to the exceptional read-out rate of these sensors (Tian, 2000).

Silicon imaging devices can usually detect wavelengths between 340 – 1100 nm (Gilblom & Yoo, 2004), the highest quantum efficiency for the camera used in this thesis (A504k, Basler AG, Ahrensburg, Germany) is between 500 and 550 nm (Micron Technology, 2004). Using a Nd:YAG DPSS laser for DPIV therefore guarantees a close to optimal light detection.

IMAGE PRE-PROCESSING

In DPIV, it is generally desirable to reduce the amount of erroneous velocity estimates to ensure the highest measurement quality possible. One common approach is the enhancement of images before the actual image correlation takes place (Raffel et al., 2007; Shavit et al., 2007). This chapter presents a selected number of promising pre-processing techniques that are implemented in `PIVlab` (see Figure 2.2 for examples).

HISTOGRAM EQUALIZATION

Contrast limited adaptive histogram equalization (CLAHE) was introduced by Pizer et al. (1987) for increasing the readability of image data in medical imaging. In the general histogram equalization, the most frequent intensities of the image histogram are spread out to the full range of the data (from 0 to 255 in 8-bit images). The advantage of CLAHE is that it does not operate on the histogram of the full image, but on small regions (tiles) of the image. This is important, because in DPIV, a uniform exposure of the whole image can often not be guaranteed (Westerweel, 1993), due to the Gaussian intensity distribution of the laser beam. In CLAHE, the contrast of every tile is optimized by histogram equalization, so that the resulting distribution matches a flat shaped histogram ranging from 0 to 255. Regions with low exposure and regions with high exposure are therefore optimized independently. After the histogram equalization, all neighbouring tiles are combined using a bilinear interpolation, resulting in an image without visible boundaries between the tiles. CLAHE significantly improves the probability of detecting valid vectors in experimental images by $4.7 \pm 3.2\%$. This improvement can be observed for a wide range of DPIV tools and correlation algorithms (Shavit et al., 2007).

INTENSITY HIGHPASS

Inhomogeneous lighting, e. g. caused by reflections from objects or inhomogeneous seeding, strongly affects the correlation signal (Raffel et al., 2007). This kind of low frequency background information can be removed by applying a high-pass filter which mostly conserves the high frequency information from the particle illumination (Gonzalez & Wintz, 1987). Such a high-pass is calculated by applying a low-pass filter to the image (blurring the image), and subtracting the result from the original image. The high-pass filter emphasizes the particle information in the image, and suppresses any low frequency information in the images (including all low frequency displacement information).

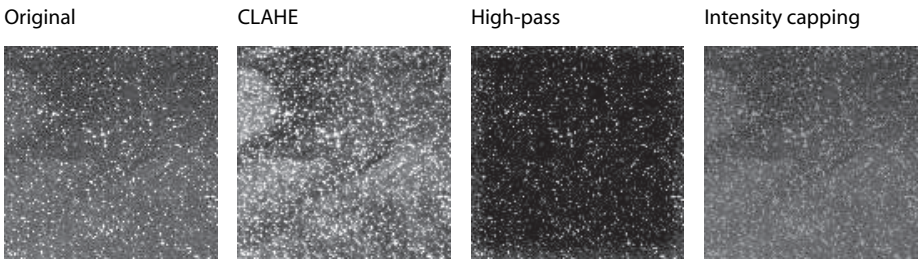

Fig. 2.2: The effect of several pre-processing techniques, see text for a description.

INTENSITY CAPPING

DPIV analyses the displacement of groups of tracer particles contained in one interrogation window statistically. The method assumes that all particles within the window have the same motion. This will not be the case in reality, as perfectly uniform flow does hardly exist. Bright particles or bright spots within the area will contribute statistically more to the correlation signal, which may bias the result in non-uniform flows (Shavit et al., 2007). The intensity capping filter proposed by Shavit et al. (2007) circumvents this problem. An upper limit of the greyscale intensity is selected, and all pixels that exceed the threshold are replaced by this upper limit. Therefore, unlike CLAHE, only a small amount of the pixel intensity information is adjusted, limiting the potential negative impact of image modifications (Shavit et al., 2007). Intensity capping improves the probability of detecting valid vectors in experimental images by $5.2 \pm 2.5\%$, and is supposed to outperform CLAHE under certain experimental circumstances (Shavit et al., 2007). Combining the intensity capping filter with the CLAHE filter does additionally increase the probability of valid vector detection in `PIVlab`. For both the CLAHE and the intensity capping technique (used separately or in combination), no significant effect on the trueness and precision of DPIV measurements could be determined.

IMAGE EVALUATION

The most sensitive part of a DPIV analysis is the cross-correlation. This part also significantly impacts the accuracy of DPIV. Small sub images (interrogation areas) of an image pair are cross-correlated to derive the most probable particle displacement in the interrogation areas. In essence, the cross-correlation is a statistical pattern matching technique that tries to find the particle pattern from interrogation area A back in interrogation area B. This statistical technique is implemented with the discrete cross-correlation function (Huang et al., 1997):

$$C(m,n) = \sum_{i} \sum_{j} A(i,j)B(i-m, j-n) \tag{2.2}$$

where A and B are corresponding interrogation areas from image A and image B.

The discrete cross-correlation function hence measures the agreement of interrogation area A and interrogation area B for a given shift (Raffel et al., 2007). The location of the intensity peak in the resulting correlation matrix C gives the most probable displacement of the particles from A to B (Huang et al., 1997).

There are two common approaches to solve equation 2.2. The most straightforward approach is to compute the correlation matrix in the spatial domain (see Figure 2.3 for a graphical representation of this correlation). This approach is either called direct cross-correlation (DCC, e. g. Okamoto et al., 2000), particle image pattern matching (PIPM, e. g. Huang et al., 1997), or convolution filtering (e. g. Stamhuis, 2006).

Another approach is to compute the correlation matrix in the frequency domain (discrete Fourier transform, DFT). The DFT is calculated using a fast Fourier transform (FFT, e. g. Soria, 1996). Both approaches have advantages as well as some drawbacks; these will be presented in short in the next sections. More details on the mathematical background of cross-correlation can be found in e. g. Keane & Adrian (1992) and Raffel et al. (2007).

DIRECT CROSS-CORRELATION (DCC)

The direct cross-correlation computes the correlation matrix in the spatial domain. In DCC, the interrogation areas A and B can have two different sizes (Stamhuis, 2006). When B is chosen twice as large as A, a particle displacement of up to half the size of A will not result in any loss of information and provide a reliable correlation matrix with low background noise (see Figure 2.3, top middle and Figure 2.4, top). DCC has been shown to create more accurate results than a standard DFT approach. Both the

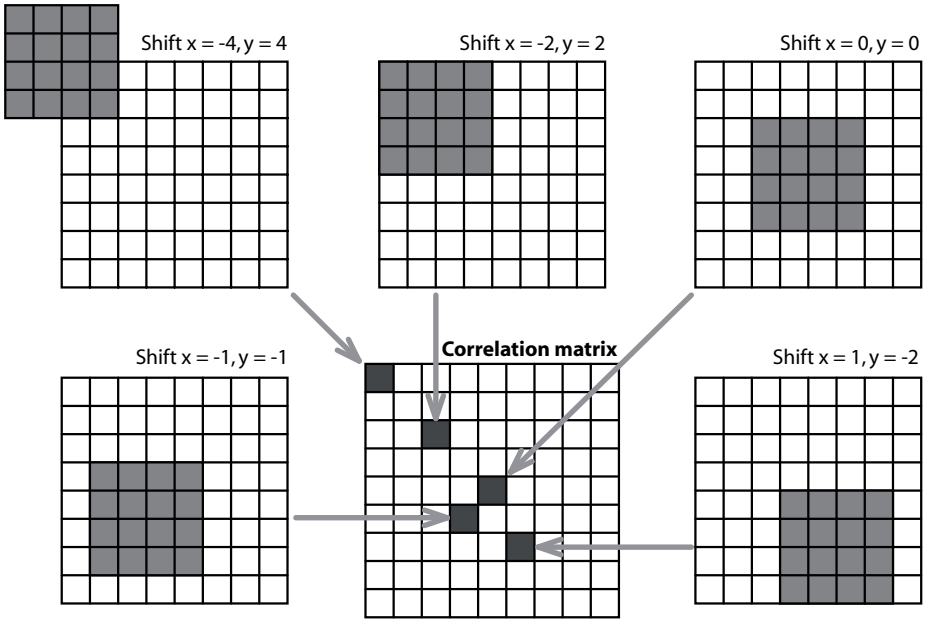

Fig. 2.3: Calculation of the correlation matrix using DCC as it is performed in MATLAB. Interrogation area A (size 4·4 pixels) is correlated with interrogation area B (size 8·8 pixels) and yields the correlation matrix (size 9·9 pixels). Adapted from Raffel et al. (2007).

systematic error and the random error of the calculation decrease substantially using DCC (Huang et al., 1997, also see the results on accuracy in the following sections). The disadvantage of DCC is the increased computational cost with respect to a standard DFT approach, especially with large interrogation areas (Soria, 1996; Huang et al., 1997; Raffel et al., 2007, see Figure 2.5A). However, recent advances in computer technology make it possible to conduct the calculations in parallel on Graphics Processing Units (GPUs), which dramatically increases the speed of the calculations (e. g. Tarashima et al., 2010), and potentially makes the DCC approach a fast and very suitable choice.

DISCRETE FOURIER TRANSFORM (DFT) AND ADVANCED DFT TECHNIQUES

The potential drawback of DCC - the computational cost - can be resolved by calculating the correlation matrix in the frequency domain (Raffel et al., 2007) using FFT (see Figure 2.5A). This approach uses interrogation areas of identical size; therefore every particle displacement induces some loss of information, which can be noticed by the increasing amount of background noise in the correlation matrix (see Figure 2.4, bottom). This

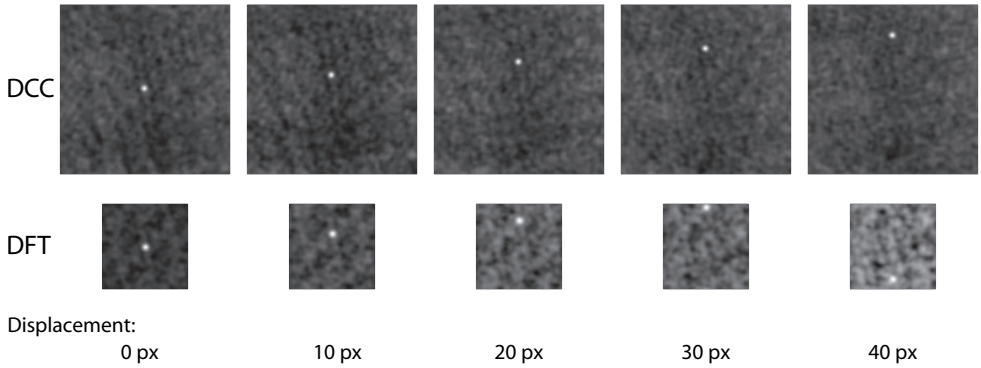

Fig. 2.4: Correlation matrices of the DCC (top) and the DFT approach (bottom), interrogation area A is 64·64 pixels for both DCC and DFT. Area B is 128·128 pixels in DCC and 64·64 pixels in DFT. In DCC, the background noise does not increase up to a displacement of 32 pixels. In DFT, background noise immediately increases if the displacement is larger than 0 pixels. A displacement of more than 32 pixels will flip the correlation peak to the opposite side of the correlation matrix, and thus makes correct measurements impossible.

background noise complicates the detection of the intensity peak and decreases accuracy. Another problem is the fact that FFT by definition assumes that the input datasets (interrogation areas) are periodic, hence that they repeat themselves in all directions (Raffel et al., 2007). As soon as the displacement of the particles is larger than half the size of interrogation area, the intensity peak in the correlation matrix is folded back into the matrix and will appear on the opposite side of the matrix (Raffel et al., 2007, see Figure 2.4, bottom right). Therefore, the displacement of the particles obligatory has to be smaller than half the size of the interrogation area. It is advised to reduce the displacement further more to about one quarter of the interrogation area, in order to keep the background noise in the correlation matrix low (Keane & Adrian, 1990). This can be achieved either by increasing the size of the interrogation windows, by decreasing Δt or by reducing the image magnification of the camera. This disadvantage can be offset by implementing 'repair routines' (Stamhuis, 2006), e. g. by running several passes of the DFT on the same dataset. Westerweel et al. (1997) have shown that the use of several DFT passes significantly increases the signal to noise ratio. In their modification of the standard DFT, the integer result of the first analysis pass is used to offset the interrogation area in the following passes. The loss of information due to particle displacement is hence minimized. Scarano & Riethmuller (1999) further enhanced this approach and proposed to refine the interrogation grid with every pass: The first pass uses large interrogation areas and can therefore accept large particle displacements. In the following passes, the area is reduced and displaced at the same time. This yields a high spatial resolution in

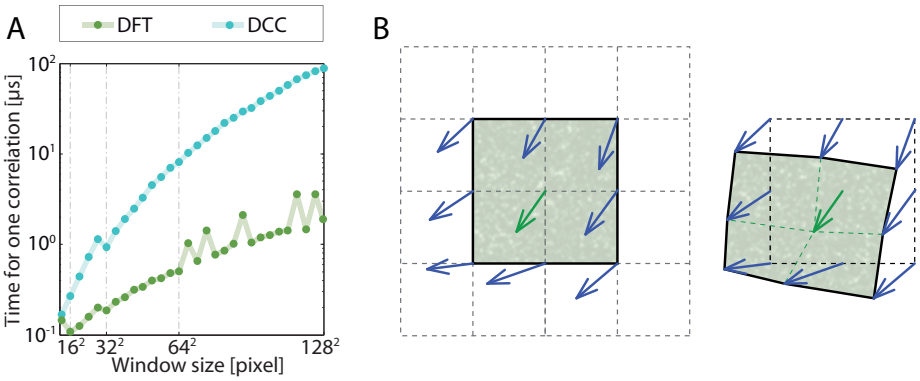

Fig. 2.5: A: Calculation speed of DCC in comparison with DFT (both calculations performed in Matlab). For the FFT calculations, FFTW is used ('Fastest Fourier Transform in the West', Frigo & Johnson, 2005) which accepts inputs of arbitrary size, but is slow for sizes that are prime or which have large prime factors (note the peaks in the graph). Generally, the DFT approach is much faster. **B:** Principle of the window deformation technique. Left: After the first interrogation pass, displacement information is present at nine positions inside the interrogation area. This information is interpolated to derive the displacement of every pixel of the interrogation area. Subsequently, interrogation area B is deformed, followed by several additional interrogation passes.

the final vector map, together with a high dynamic velocity range and it increases the signal to noise ratio.

So far, it was assumed that the particles within the interrogation area have a uniform motion. This is hardly the case in real flows; the particle patterns will additionally be sheared and rotated. Non uniform particle motion within the interrogation area will broaden the intensity peak in the correlation matrix and deteriorate the result. Several methods that account for the transformation of the interrogation areas have been proposed (e. g. Huang et al., 1993b; Jambunathan et al., 1995; Scarano & Riethmuller, 2000). These methods can be termed 'multiple pass, grid refinement, window deformation technique', and in `PIVlab` the following procedure is implemented: The analysis is started with a regular DFT analysis. The first pass yields displacement information at the centre of each interrogation area. When the areas overlap one another by e. g. 50%, there is additional displacement information at the borders and corners of each interrogation area (nine positions in total, see Figure 2.5B, left). This information is used to calculate displacement information at every pixel of the interrogation areas via bilinear interpolation. Next, interrogation area B is deformed according to this displacement information (see Figure 2.5B, right) using either bilinear interpolation (faster) or spline interpolation (higher precision, but slower). The next interrogation pass correlates the original interrogation area A with the deformed area B. The remaining displacement

information of each pass is accumulated. After a few passes, the deformed interrogation area B will look almost identical to the original area A, and the displacement has been determined with high accuracy. Between the passes, but not after the final pass, the velocity information is smoothed and validated and missing information is interpolated. Data validation can be relatively strict, as any deteriorating effect of interpolation and smoothing will be corrected in the correlation of the following pass.

PEAK FINDING

The choice of the peak finding technique is - similar to the choice of the cross-correlation technique - another important factor for the accuracy of DPIV. The integer displacement of two interrogation areas can be determined straightforward from the location of the intensity peak of the correlation matrix. The location can be refined with subpixel precision using a range of methods (e. g. Lourenco & Krothapalli, 1995; Roesgen, 2003; Raffel et al., 2007). Several of the existing subpixel estimation techniques were tested in `PIVlab`, the two most promising methods in terms of accuracy, speed and universality were implemented. When the integer peak location in the correlation matrix is known, a Gaussian function can be fitted to the intensity distribution (see Figure 2.6). It is sufficient to use only the directly adjacent vertical and horizontal pixels (two times a 3-point fit = 2 · 3-point fit) and to evaluate the x and y axis separately. A Gaussian function is an appropriate candidate for the fit, as the individual particle images closely match a Gaussian intensity distribution, and the cross-correlation of two Gaussian distributions again yield a correlation matrix with a Gaussian distribution (Lourenco & Krothapalli, 1995). The peak of the fitted function is used to determine the particle displacement with subpixel precision. This kind of subpixel estimator works very well in practice (e. g. Forliti et al., 2000) and is the standard in DPIV processing (Nobach & Honkanen, 2005). If the particle displacement within an interrogation area is exposed to shear or rotation or if the images suffer from excessive motion blur, the displacement peak may have an elliptical shape (Raffel et al., 2007). In this case, the Gaussian 2 · 3-point subpixel function can cause bias error. This can be avoided by fitting a two-dimensional Gaussian function (9-point fit) as proposed by Nobach & Honkanen (2005). However, the application of the window deformation technique introduced earlier in this chapter reduces shear and rotation within the interrogation area. Therefore, the added value of using a two-dimensional Gaussian function is more pronounced in non-deforming methods, such as DCC and single pass DFT. In `PIVlab`, both the 2 · 3-point and the 9-point Gaussian fit are implemented.

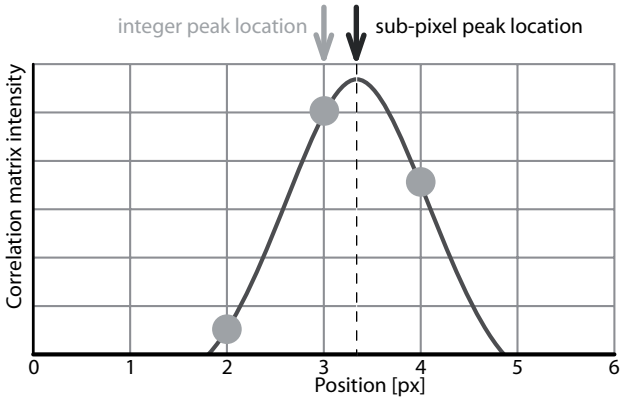

Fig. 2.6: Principle of the Gaussian 2·3-point fit: Subpixel precision is achieved by fitting a one-dimensional Gaussian function (solid line) to the integer intensity distribution of the correlation matrix (dots) for both axes independently (only one axis is shown here).

EVALUATION OF THE ACCURACY

The quality of a measurement is determined by the magnitude of error included in the measurement. In DPIV, two main sources of error exist: The bias error (ε_{bias}) and the random error (ε_{rms}) both contribute to the total error of a measurement (Raffel et al., 2007). The bias error determines the trueness of displacement estimates. Trueness is defined as the agreement between the mean result of a large series of measurements and the true displacement (Menditto et al., 2007). The random error determines the precision of the displacement estimate. Precision is a measure for the spread of displacement estimates (Menditto et al., 2007). It may well be that a measurement is very precise, but the mean value of the measurements is not correct because trueness is low (see Figure 2.7, top right). Together, trueness and precision determine the accuracy of a DPIV system. For the determination of the bias and the random error, large numbers of measurements (Monte Carlo simulation) have to be performed for getting statistically relevant results. Additionally, the precise difference between displacement estimates and the true displacement has to be known. The most convenient way to satisfy these requirements is to use synthetic particle images. The synthetic images were generated in `PIVlab` under known conditions following Raffel et al. (2007): The particles have a Gaussian intensity profile with known diameter. A known amount of particles is placed at random positions within the simulated laser sheet having a Gaussian intensity profile. Simulation parameters may be varied to study their effect on accuracy. The bias error is calculated following:

$$\varepsilon_{bias} = \frac{1}{n}\sum_{i=1}^{n} d_{meas,i} - d \tag{2.3}$$

where d_{meas} is the displacement measured by a specific DPIV algorithm, d is the true displacement given by the particle image generator.

The random error is determined as:

$$\varepsilon_{rms} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\overline{d_{meas}} - d_{meas,i})^2} \tag{2.4}$$

where $\overline{d_{meas}}$ is the mean measured displacement (see Figure 2.8 for an example). Here, the accuracy of a selection of DPIV algorithms with respect to several experimental situations is presented (see Table 2.1). All simulations were performed with a sample size of $n \geqslant 9.3 \cdot 10^5$.

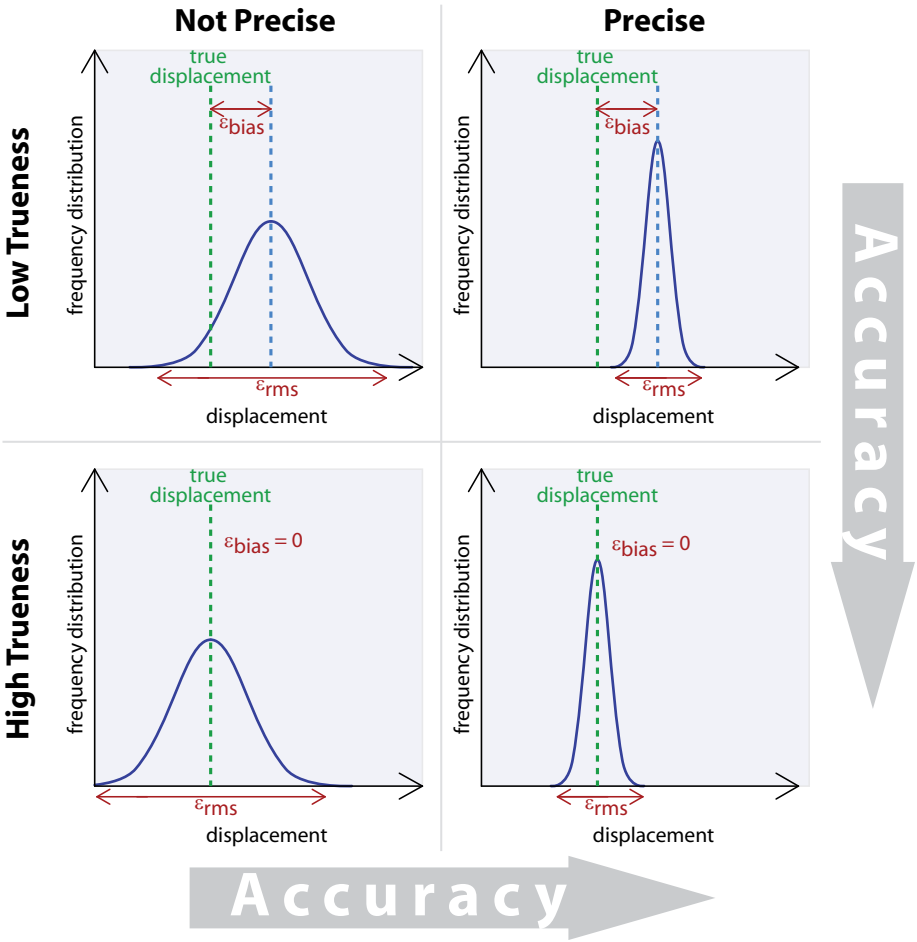

Fig. 2.7: Definition of trueness, precision and accuracy for DPIV systems.

Bias error is caused by the peak-locking effect which exists in most cross-correlation algorithms. The error is mostly produced by fitting a smooth curve through the integer intensity distribution of the correlation matrix during peak finding (Chen & Katz, 2005). The bias error is a function of the displacement; the magnitude depends mainly on the correlation algorithm, the subpixel estimation technique and on the particle diameter. The calculated subpixel displacement is biased towards integer values and the effect becomes worse for small particle image diameters. Particle image diameters below 3 pixels result in a very narrow intensity peak in the correlation matrix, which can not sufficiently be approximated by a Gaussian fit. This can clearly be observed in Figure 2.9: Here, all of the DPIV algorithms tested have a significant lack of accuracy because all rely on the same $2 \cdot 3$-point subpixel estimator that is not suitable for very small particle image

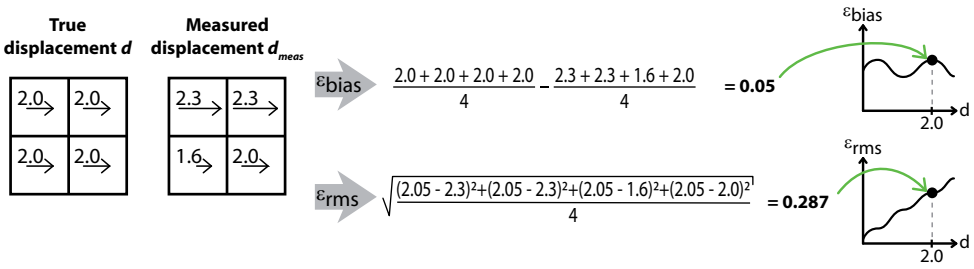

Fig. 2.8: Calculation of the bias error ε_{bias} and the random error ε_{rms} from the true displacements d and the measured displacements d_{meas} in a $2 \cdot 2$ grid ($\Rightarrow n = 4$). This example also shows how the graphs that result from the accuracy tests can be read.

diameters. The error decreases substantially for particle image diameters of 3 pixels (see Figure 2.10) and for 5 pixels (see Figure 2.11). The basic DFT algorithm has the poorest performance.

Tab. 2.1: DPIV algorithms under test

Parameter set	**'DCC 1'**	**'DFT 1'**	**'DFT multi lin'**	**'DFT multi spl'**
CLAHE, size [px]	50	50	50	50
DPIV Technique	DCC	DFT	DFT	DFT
Subpixel estimator	Gaussian 2 · 3-point	Gaussian 2 · 3-point	Gaussian 2 · 3-point	Gaussian 2 · 3-point
Window deformation	none	none	linear	spline
Overlap[%]	50	50	50	50
Pass 1 int. area [px]	16 · 16	16 · 16	64 · 64	16 · 16
Pass 2 int. area [px]	-	-	32 · 32	16 · 16
Pass 3 int. area [px]	-	-	16 · 16	16 · 16
Pass 4 int. area [px]	-	-	-	16 · 16

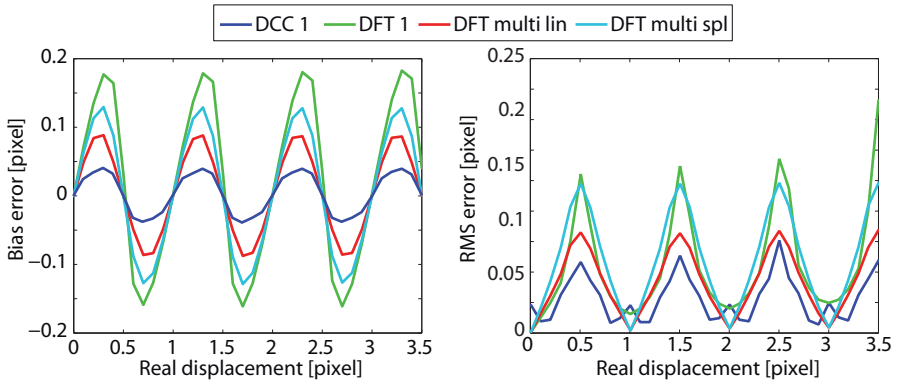

Fig. 2.9: Bias error and random error for particle images with a diameter of 1 pixel.

PARTICLE DIAMETER

The DPIV experiments presented in this thesis use particles with a diameter of 57 µm. These particles appear in the final images with a mean diameter of 3.8 ± 1.6 pixels ($n = 1.8 \cdot 10^6$). The particle size in DPIV experiments generally varies (Huang et al., 1997), and influences the accuracy of the analyses. Guezennec & Kiritsis (1990) found that the accuracy decreases with particle size for particles with an image diameter larger than 2 pixels. More detailed studies identified the existence of an optimal particle image diameter of about 1.5 pixels (Raffel et al., 2007). The influence of the particle image diameter on the accuracy of the analysis depends on the DPIV algorithm used; therefore the effect of particle image diameter has to be studied in `PIVlab`. The results (see Figure 2.12B) show that the random error has a local minimum at a diameter of about 1.8 to 2

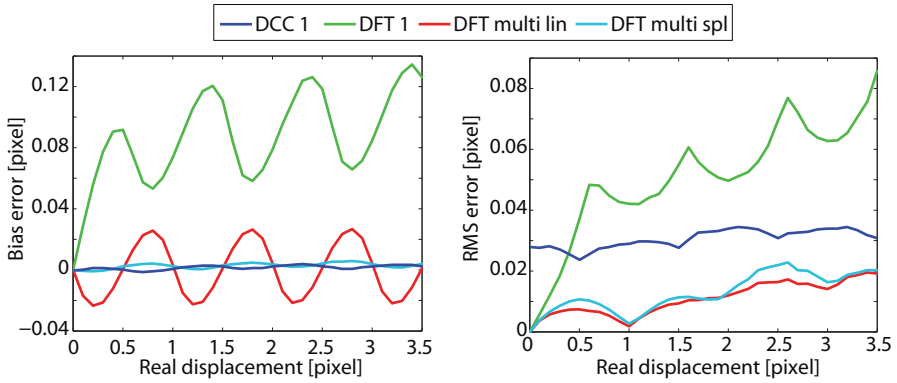

Fig. 2.10: Bias error and random error for particle images with a diameter of 3 pixels.

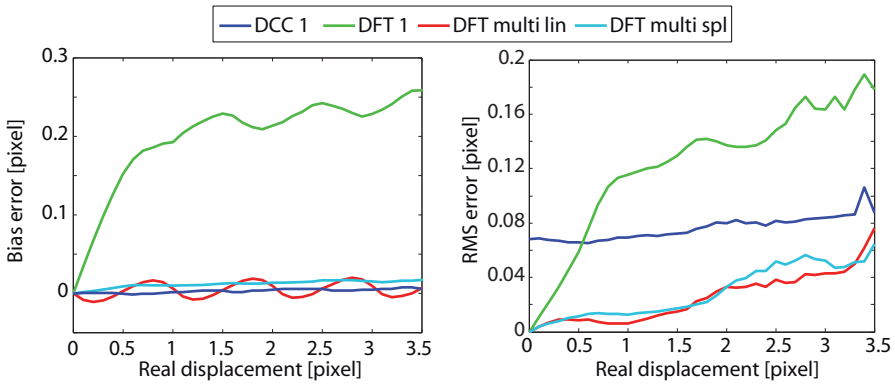

Fig. 2.11: Bias error and random error for particle images with a diameter of 5 pixels.

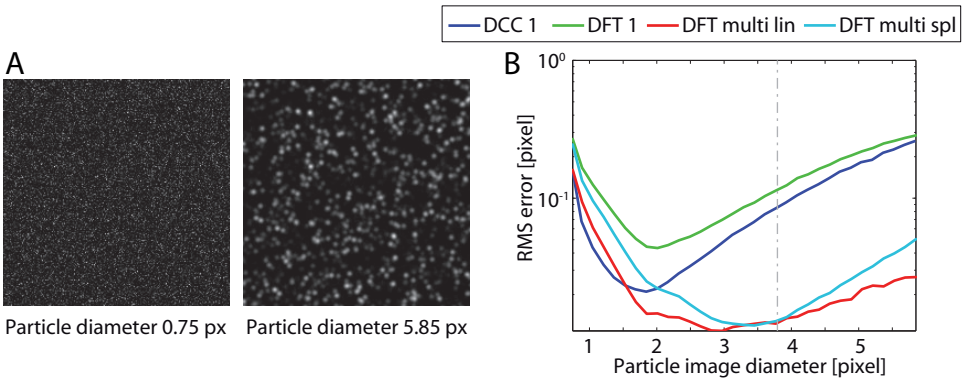

Fig. 2.12: A: Example of the particle image diameters tested. **B:** The effect of particle image diameter on the random error (particle displacement = 3.5 pixels).

pixels for basic DCC and DFT (see Figure 2.12A for an example of the range of diameters tested). The minimum random error is shifted towards slightly larger particle image diameters for window deformation techniques (see Figure 2.12B). Here, the window deformation algorithms also have a significantly smaller random error than the basic DCC and DFT approach (note the logarithmic scaling of the y-axis). The particle image diameter used in this thesis (note the dashed line in Figure 2.12B) is confirmed to be in the optimal range of the window deformation algorithms for highly accurate measurements.

PARTICLE DENSITY

The density of particles within an interrogation area is important for the quality of an analysis. Keane & Adrian (1992) report that a valid vector detection probability of more

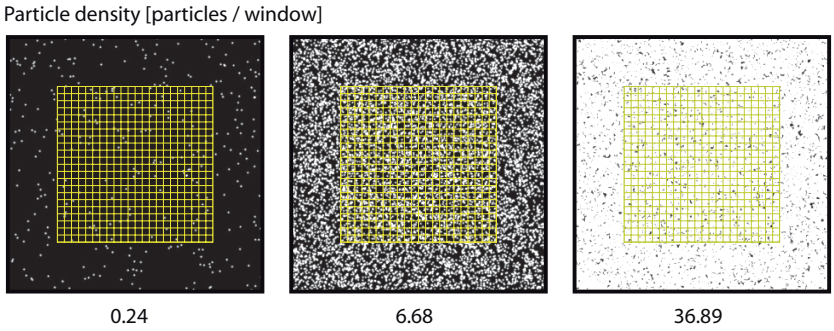

Fig. 2.13: Example of the particle densities tested. The yellow grid represents the interrogation areas of 16·16 pixels.

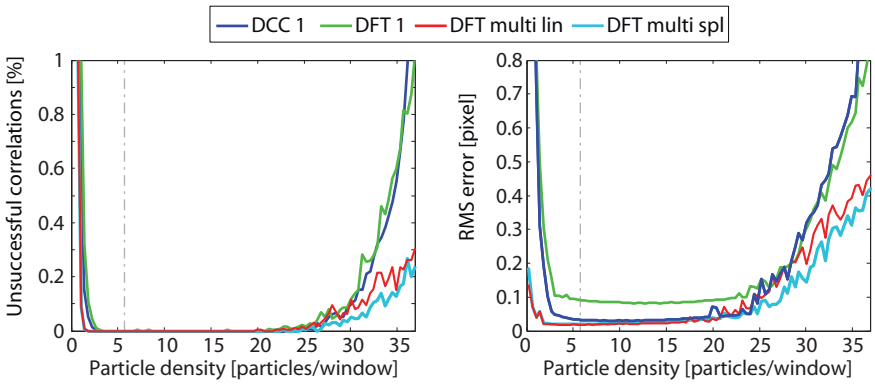

Fig. 2.14: The effect of particle density on the rate of unsuccessful correlations and on the random error (particle image diameter = 3 pixels, displacement = 3.5 pixels).

than 95% can be achieved, if the particle density is larger than 5 particles per interrogation area. Furthermore, the random error is reported to decrease substantially with increasing particle density (Huang et al., 1997). The results that are reported in the current chapter indicate that this is not always the case (see Figure 2.13 for an example of the densities tested). Particle densities below 3 particles per interrogation area decrease the amount of successful correlations (defined as a correlation with a clear peak and with a displacement estimate that deviates less than 1 pixel from the true displacement), and increase the random error (see Figure 2.14). All DPIV algorithms can accept up to about 20 particles per interrogation area before the amount of unsuccessful correlations increases. The maximum amount of unsuccessful correlations within the range of particle densities tested is however very small ($\leqslant$ 1%). The mean particle density in the DPIV experiments presented in this thesis is 5.80 $\pm$ 0.48 particles per interrogation area ($n = 7.4 \cdot 10^3$). The selected particle density is within the optimal range (note the grey line in Figure 2.14).

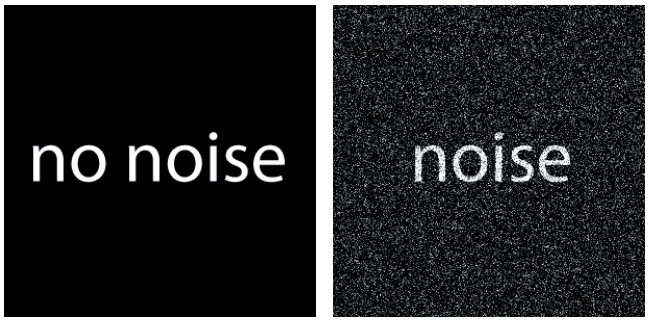

Fig. 2.15: Example of the noise levels tested.

SENSOR NOISE

It is known that image noise influences DPIV analyses (e. g. Guezennec & Kiritsis, 1990; Huang et al., 1997; Raffel et al., 2007). The noise in DPIV images is caused during the conversion of photons into electric current within the image sensor. Under challenging lighting conditions, CMOS image sensors suffer from temporal noise. This is caused by the thermal noise of the highly integrated circuitry in the image sensor (Tian, 2000). For the analysis of the effect of noise on the rate of successful correlations and on the random error, the noise is approximated by simulating Gaussian white noise (see Figure 2.15 for an example of the range of noise tested). Window deformation and the basic DCC algorithms can deal with large amounts of noise before correlations fail (see Figure 2.16). The basic DFT algorithm fails earlier and also shows a higher random error than the window deformation algorithms. The amount of image sensor noise in the DPIV experiments presented in this thesis can hardly be quantified; visual inspection of the images implies a noise level with a variance somewhere below 0.01, which will hardly degrade the analyses.

PARTICLE PAIR LOSS

This thesis includes measurements of the aerodynamics of flapping wings that typically generate highly three-dimensional flows (e. g. David et al., 2012). Velocity components normal to the laser sheet induce a loss of information, as some particle pairs leave the measurement volume and therefore reduce the correlation between two successive images. This reduces the probability of valid displacement estimates (Raffel et al., 2007). The effect of out-of-plane motion was simulated by removing random particles from a synthetic image pair and introducing the same amount of new particles at random positions. The analysis allows to estimate the sensitivity of different DPIV algorithms to out-of-plane motion. The basic DFT approach is relatively sensitive to particle pair loss

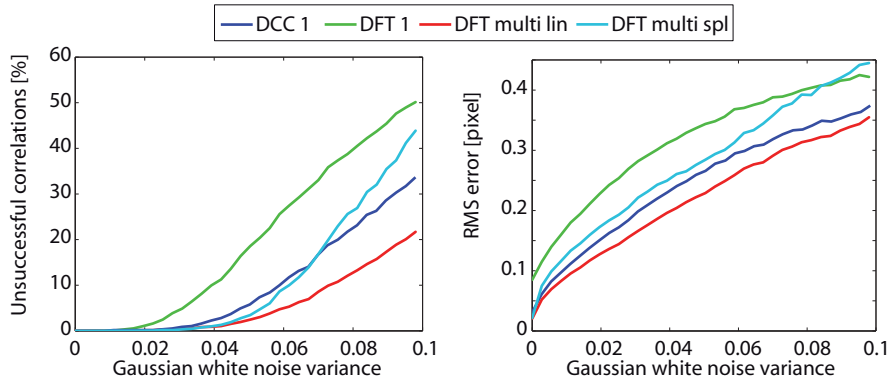

Fig. 2.16: The effect of Gaussian white noise on the rate of unsuccessful correlations and on the random error (particle image diameter = 3 pixels, displacement = 3.5 pixels).

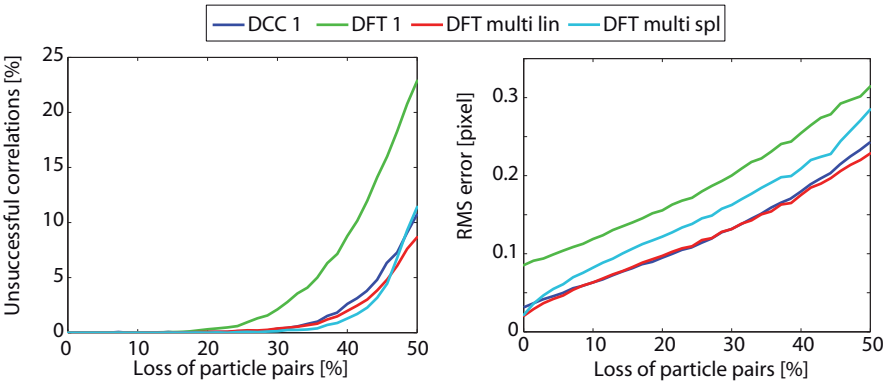

Fig. 2.17: The effect of out-of-plane motion on the rate of unsuccessful correlations and on the random error (particle image diameter = 3 pixels, displacement = 3.5 pixels).

(see Figure 2.17). When 10% of the particles are replaced, some unsuccessful correlations can already be observed. Window deformation algorithms are more robust, and also show a slightly smaller random error. The result confirms the adequacy of window deformation algorithms when analyzing highly three-dimensional flow phenomena.

MOTION BLUR

In the DPIV studies on flapping wings (see chapters III and IV), the most limiting factors were the amount of light delivered from the laser and the light sensitivity of the camera. To capture enough light for reliable DPIV analyses, the exposure time of the camera had to be

Fig. 2.18: Example of the level of motion blur tested.

set to a relatively high value. The particles moved during the exposure, hence the images show a certain amount of motion blur, and it is likely, that such blurred particle images influence the quality of DPIV analyses: Motion blur distorts the shape of the correlation peaks, and this hampers the estimation of subpixel accurate displacements (Nobach & Honkanen, 2005). The effect of motion blur was simulated by convolving the images with a one-dimensional filter of the desired length with equally distributed weights (see Figure 2.18 for an example of the effect of motion blur). Window deformation algorithms are significantly less affected by the presence of motion blur than basic DFT and DCC algorithms (see Figure 2.19A). The amount of motion blur in the experimental images is estimated to correspond to a point-spread function length of 4.9 pixels (exposure time = 1500 µs; mean flow velocity = 3250 pixels/s, the average displacement during exposure is therefore $1.5 \cdot 10^{-3}$ s $\cdot$ 3250 px/s $\approx$ 4.9 px, see dashed line in Figure 2.19A). Although motion blur does slightly degrade the precision of a measurement, window deformation is a suitable choice for limiting the negative impact.

PROCESSING SPEED

As mentioned in short earlier in this chapter, a DFT is computationally more efficient than a direct computation of the correlation matrix using DCC. However, the use of 'repair routines' introduces significant additional computational load and makes the advanced DFT slower than the DCC (see Figure 2.19B). This disadvantage is however offset by the large increase in accuracy and robustness of window deformation techniques, as has been shown in this chapter.

SHEAR

In the introduction about window deformation techniques, the deteriorating effect of shear within the interrogation area was already noted. Window deformation limits the

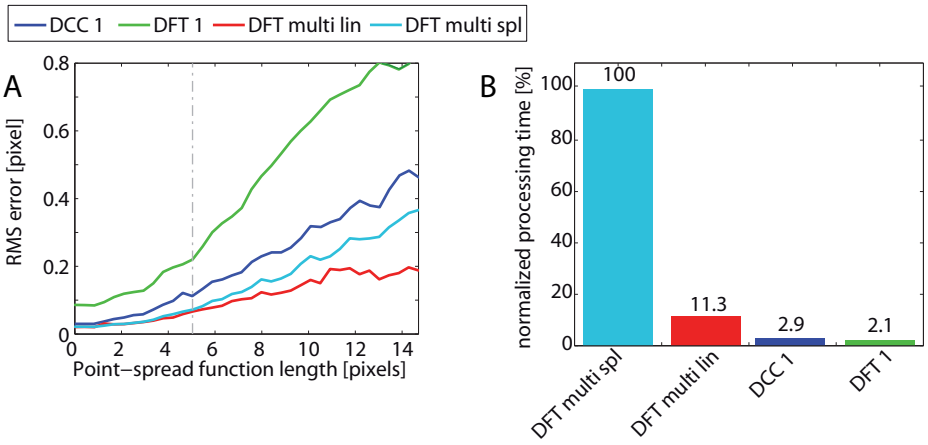

Fig. 2.19: A: The effect of motion blur on the random error (particle image diameter = 3 pixels, displacement = 3.5 pixels). **B:** Normalized processing time of the algorithms tested in this chapter.

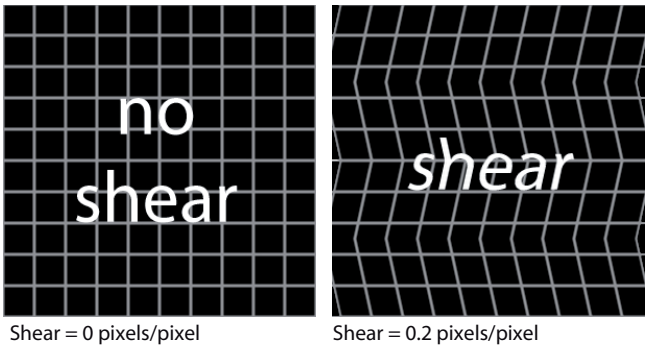

Fig. 2.20: Example of the magnitude of shear tested.

negative impact of shear on the accuracy (Huang et al., 1993a; Fincham & Delerce, 2000). Shear was simulated by applying a triangle wave displacement to the synthetic particle images (see Figure 2.20 for an example of the effect). The simulation shows, that the basic DFT is sensitive to shear as the bias error increases substantially (see Figure 2.21). The window deformation technique using a bilinear interpolation for deforming the images largely increases the accuracy. DCC is much less prone to shear than DFT. The highest accuracy in terms of bias and random error is achieved by the DFT window deformation technique that uses spline interpolation.

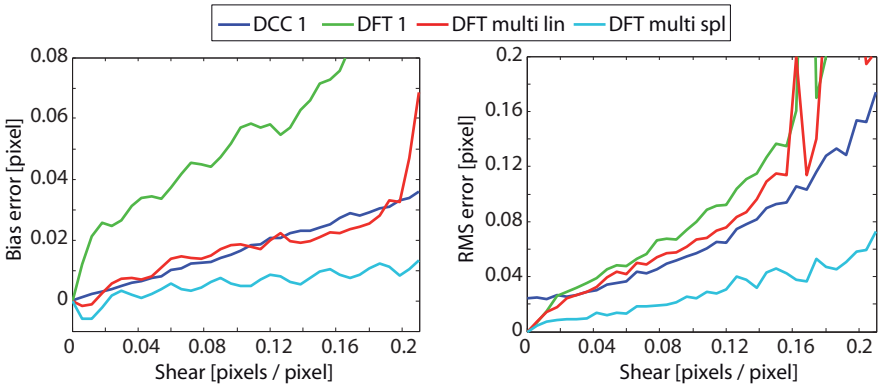

Fig. 2.21: The effect of shear on the bias error and the random error (particle image diameter = 3 pixels, triangle wave displacement).

EXPERIMENTAL IMAGES

In the accuracy tests in the preceding sections, only one parameter was varied at a time. This procedure enables to determine the sensitivity of DPIV algorithms with respect to different parameters, and to optimize the experimental setup accordingly. In real DPIV experiments, the images will always suffer from a combination of all the properties under test: The effects of noise, particle density, particle diameter, out-of-plane motion, blur and shear will all coincide in these experiments. Analyses of experimental images show, that the window deformation techniques are much more robust under highly challenging conditions (see Figure 2.22). The analyses of the experimental images of a flapping wing show that DCC suffers from 13.4 $\pm$ 9.6 times more missing correlations under challenging conditions (n = 176, see Figure 2.23). Therefore, it is advantageous to use window deformation algorithms whenever the experimental conditions are suboptimal and a short processing time is not the main interest.

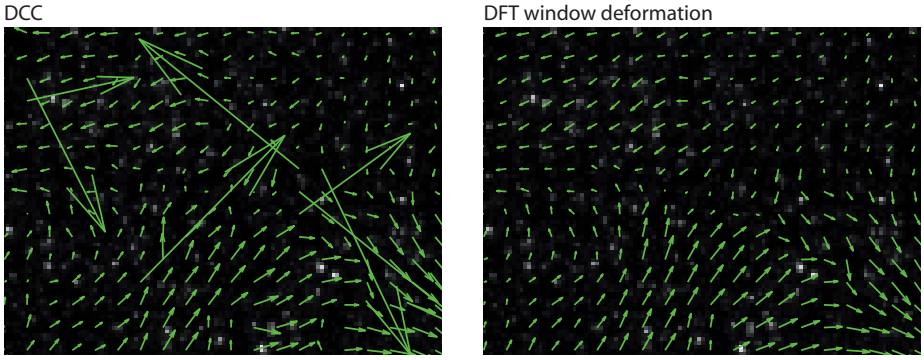

Fig. 2.22: Comparison of the velocity map calculated using DCC (left) and window deformation (right) under challenging experimental conditions. Large vectors are outliers. The window deformation technique reduces the amount of erroneous correlations.

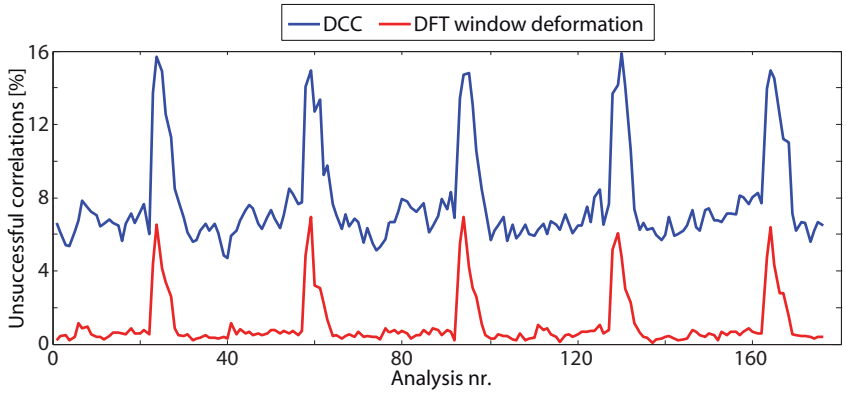

Fig. 2.23: Preliminary analyses of DPIV images with very low lighting of a flapping wing. The amount of unsuccessful correlations is considerably lower when using window deformation (the periodic peaks result from some reflections of the flapping wing entering the field of view).

POST-PROCESSING

DATA VALIDATION

Although the advanced cross-correlation algorithms provide very robust velocity estimates within the interrogation areas, bad lighting conditions, strong three-dimensional flow or reflections from objects within the light sheet still cause a certain amount of erroneous correlations. Post processing of DPIV data is generally required to obtain reliable results (Nogueira et al., 1997). The dataset of three-dimensional flow studies, like the ones presented in this thesis, readily contains more than $2.5 \cdot 10^8$ UVW vectors. In these experiments, the flapping wing obscures many interrogation areas and sometimes reflects the light of the laser. The number of erroneous vectors (outliers) will therefore be considerable, even when using the most advanced image correlation techniques. It is hence necessary to find methods of (semi-) automatic data validation in order to effectively suppress erroneous data. A very basic method to filter outliers is to choose thresholds for acceptable velocities (e. g. via graphical selection, see Figure 2.24). This method has to be used with caution, as it is based on the experience of the experimenter, and therefore has very subjective properties. In the example shown in Figure 2.24, the

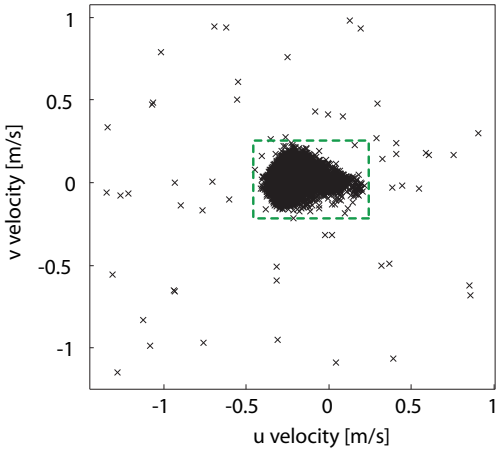

Fig. 2.24: Scatter plot of u and v velocities in the DPIV analysis of a van Kármán wake. Data inside the green rectangle in the centre is supposed to be valid, vectors outside of the rectangle are supposed to be outliers (0.42% of the total data in this example).

choice of the velocity thresholds is based on assumptions on realistic flow velocities in the test section. Velocity thresholds can also be determined semi automatically. Each velocity component can be compared with a lower threshold and an upper threshold (t_{lower} and t_{upper}):

$$t_{lower} = \overline{u} - n * \sigma_u \tag{2.5}$$

$$t_{upper} = \overline{u} + n * \sigma_u \tag{2.6}$$

where $\overline{u}$ = mean velocity; σ_u = standard deviation of u.

The user defined constant n determines the strictness of this filter. Consequently, velocities that are not in between these thresholds will be rejected. This simple filter works very well in practice, as it adapts to some extent to the nature of the flow: In very turbulent flow, the standard deviation will be high, and the filter is less strict. In laminar flow situations, the standard deviation is likely to be small, and the filter will tend to reject vectors that deviate little from the mean velocity.

The data validation techniques described above both require an experienced experimenter - a prerequisite that is unfortunately not always available. A more universal outlier detection method that automatically adapts to local flow situations was proposed by Westerweel & Scarano (2005): The normalized median test (or local median filter), evaluates the velocity fluctuation with respect to the median in a $3 \cdot 3$ neighbourhood around a central vector. The median of this fluctuation is then used as normalization for a more classical median test. The universality of this test was demonstrated by Westerweel & Scarano (2005) for a range of different situations. `PIVlab` features all three of these validation techniques, as the total quality of vector validation can be improved by the combination of several techniques.

DATA INTERPOLATION

After the removal of outliers, missing vectors should be replaced by interpolated data (Nogueira et al., 1997). The interpolation is recommended in all cases, even when it is desired not to modify the original dataset: Missing vectors become increasingly problematic when derivatives are calculated. Single missing vectors may leave large gaps in the dataset due to the numerical nature of the derivative calculation (e. g. forward difference or central differentiation scheme). Common interpolation techniques all rely on the data surrounding the missing vector(s) to fill the gap. One common technique is the $3 \cdot 3$ neighbourhood ($3 \cdot 3$ mean) interpolation. A two-dimensional linear interpolation is another alternative. Stamhuis & Videler (1995) propose to use a two-dimensional spline interpolation to derive missing vectors. `PIVlab` uses a boundary value solver for interpolation, which was originally developed for reconstructing images with missing information. The approach provides an interpolation that is generally fairly smooth, and over larger regions with missing data, it will tend towards the average of the boundary velocities, which prevents overshooting (D'Errico, 2012). The performance of the most

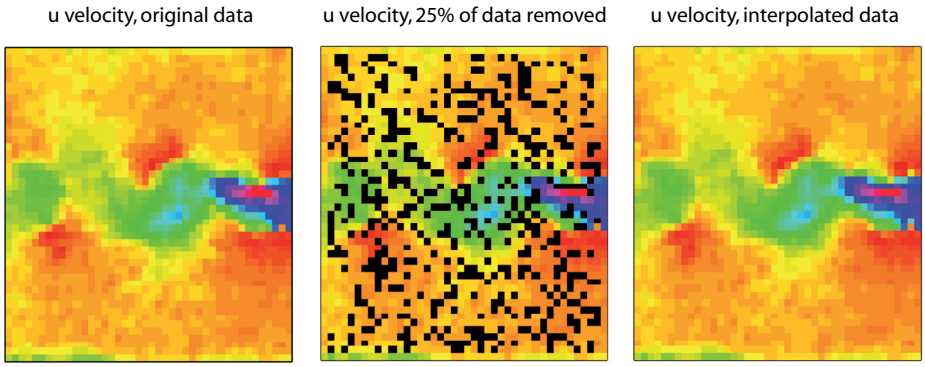

Fig. 2.25: Procedure for testing several interpolation techniques. Left: Original velocity data. Middle: Data is removed at random positions. Right: Gaps are filled with interpolation and compared to the original velocity data.

popular interpolation techniques and the boundary value solver are tested: A number of real and synthetic image pairs are analyzed using standard DPIV (see Figure 2.25, left). An increasing amount of random vectors (0% to 15%) is removed from the resulting vector matrix (see Figure 2.25, middle). The missing data is interpolated using one of the interpolators (see Figure 2.25, right), and finally, the mean absolute difference between the original data and the interpolated data is determined. This whole procedure is repeated 1000 times for each image pair and each level of missing data to get statistically relevant results.

The most basic interpolation scheme – the $3 \cdot 3$ mean neighbourhood interpolation – performs worst (see Figure 2.26). A two-dimensional spline interpolation gives very good results if the amount of missing data is below 5%. But if the amount of missing data increases, and hence the probability and size of connected missing data 'islands' increases, the spline interpolation performs worse than the boundary value solver, because splines with too few nodes easily overshoot. Between 1% and 5% of data may typically be missing in high quality DPIV data under rather optimal conditions (Raffel et al., 2007). According to the results (see Figure 2.26), a two-dimensional spline interpolation might therefore be perfectly adequate. But when analyzing biogenic flows, optimal conditions are sometimes hard to achieve: The seeding and the intensity of the laser light must be reduced in order not to harm any organisms generating the flow (Stamhuis, 2006). Challenging conditions will increase the amount of missing data; the boundary value solver is hence the more universal interpolator.

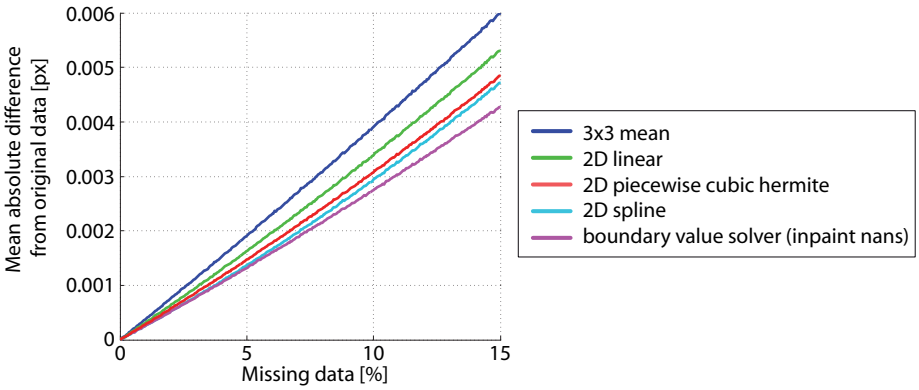

Fig. 2.26: Performance of popular interpolators. The boundary value solver performs best under the presence of larger amounts (> 10%) of missing data.

DATA SMOOTHING

As has been shown, the cross-correlation of DPIV data is never perfectly precise. A certain amount of measurement noise will be inevitable (Raffel et al., 2007). Noise is often not problematic when integral quantities – such as integrals of velocities to derive the circulation of a vortex – are desired (see Figure 2.27A, bottom). But when differentials, e. g. vorticity / shear / strain / vortex locators are used to describe a flow field, measurement noise quickly becomes a dominating quantity (Nogueira et al., 1997, see Figure 2.27A, middle). Noise can be effectively reduced by applying data smoothing. Raffel et al. (2007) propose to perform a convolution of the data with a $2 \cdot 2$ or $3 \cdot 3$ kernel with equal weights. Another common and effective method to smooth DPIV data is median filtering. More advanced smoothing algorithms are based on a penalized least squares method ('Smoothn', Garcia, 2010). The performance of the smoothing algorithms is tested using DPIV data of synthetic particle images. Two Hamel-Oseen vortices are placed in random locations in a $1024 \cdot 1024$ pixel image. The artificial flow field hence consists of regions with very high velocity gradients (in the vicinity of the vortex cores), and regions with almost constant velocity. The synthetic images are analyzed, followed by the application of one of the three smoothing algorithms ($3 \cdot 3$ mean; $3 \cdot 3$ median; 'Smoothn'). Finally, the mean absolute difference and the maximum absolute difference between the true velocities and the calculated velocities are determined. The test is run with 250 synthetic image pairs for each smoothing algorithm. An exemplary DPIV analysis of such an image pair is shown in Figure 2.27B.

Data smoothing – no matter what algorithm is used – always decreases the amount of noise introduced by the DPIV algorithm and hence increases the quality of the velocity estimation (see Figure 2.28). The application of the 'Smoothn' algorithm reduces the difference between analysis and true velocities. It was therefore decided to implement

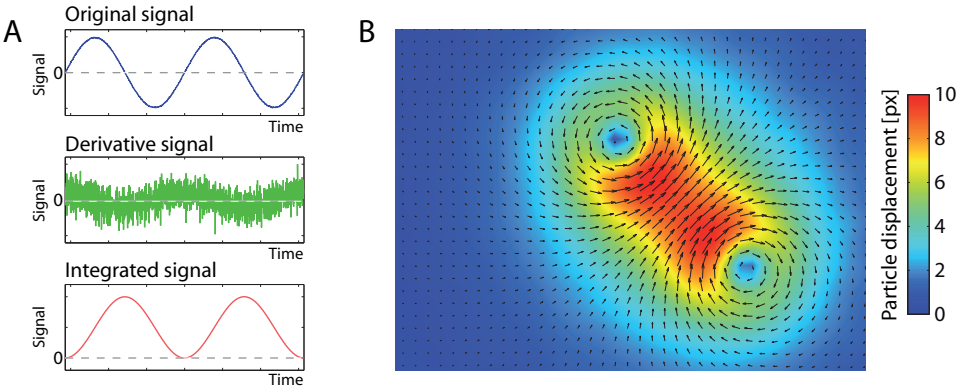

Fig. 2.27: A: The effect of measurement noise. Top: Original signal, sine wave with random noise. Middle: Derived signal, noise dominates. Bottom: Integrated signal, noise is reduced drastically. All data normalized. **B:** Exemplary DPIV result for smoothing algorithm validation. Synthetic image ($1024 \cdot 1024$ pixels), containing two randomly positioned vortex cores with a maximum particle displacement of 10 pixels. Colours represent particle displacement.

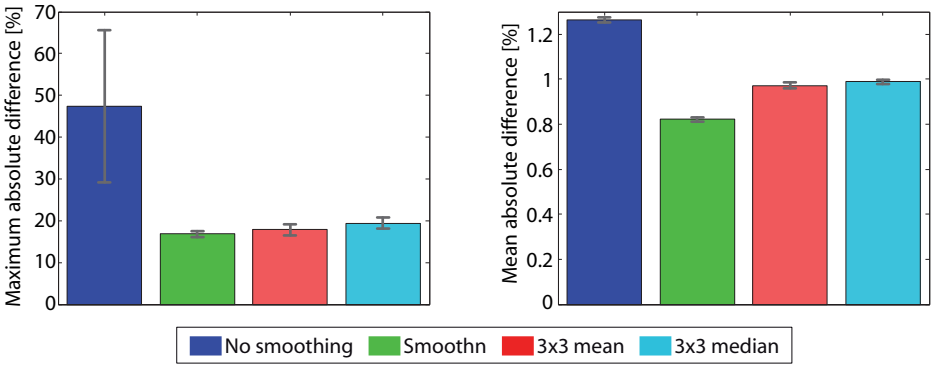

Fig. 2.28: Validation of smoothing algorithms. Left: Maximum absolute difference between the calculated velocities and the true velocities in percent of the maximum true velocity. Right: Mean absolute difference. $n = 250$

this algorithm in `PIVlab` in order to offer an option to further improve the quality of the velocity estimation.

FURTHER DATA PROCESSING

Many DPIV studies reveal relatively complex flow patterns. Such a complexity is hard to describe purely with vector maps, therefore it is necessary to further process and distil the results. This section will give a brief overview of common derivatives and their relevance. The derivative most often used to describe a flow field is vorticity (for the mathematical background, see e. g. Stamhuis & Videler, 1995). Vorticity is often used to visualize vortices. Theoretically, in the core of a vortex (solid body rotation), the magnitude of vorticity is larger than zero, outside the vortex core (irrotational region), vorticity equals zero. Care has to be taken when using it as exclusive criterion for the presence of vortices, as vorticity is also sensitive to shear. Better suited for vortex detection is the discriminant for complex eigenvalues (vortex locator / DCEV, see e. g. Stamhuis & Videler, 1995). Further derivatives useful to describe a flow pattern are divergence (a measure for out-of-plane flow in two-dimensional studies in incompressible fluids, e. g. Stamhuis & Nauwelaerts, 2005), shear rate and strain rate (see e. g. Stamhuis & Videler, 1995).

The dominant flow patterns (e. g. vortices) are often superimposed by other fluid motions. When the superimposed flow has a constant velocity, the average flow is usually subtracted to get a better view on vortices. If the velocity of the superimposed flow is not constant and if it varies locally, it can be useful to high-pass the vector field to produce a comprehensive visualization. This will remove low frequency flow patterns, and highlight local velocity gradients (see Figure 2.29A and B). Qualitative flow visualizations can also benefit from the application of line integral convolution (LIC). The approach was introduced by Cabral & Leedom (1993); it is based on a curvilinear filtering that locally blurs random noise textures. The filter is capable of rendering detail on very complex vector fields. This kind of visualization (see Figure 2.29C) is in principle comparable to releasing streamlines at every grid point of the vector field. One widespread criterion for a vortex is that streamlines follow a circular pattern around the core (Robinson et al., 1989), therefore LIC can also be used to assist the identification of vortices, velocity sources and sinks.

DPIV creates a spatially resolved vector map displaying the velocities detected in a two-dimensional cross-section through a flow field. As mentioned earlier, the complex nature of some flow patterns can be better visualized using derived parameters. Another approach to reduce complexity is to extract data from the two-dimensional velocity map using a one-dimensional path, or to focus on integral quantities (Raffel et al., 2007). One-dimensional data extraction can provide insight into the velocity and vorticity profile of vortex rings (see Figure 2.30A-C), and test whether the morphology of the vortex ring complies with theory (e. g. Spedding et al., 1984). Furthermore, by determining the

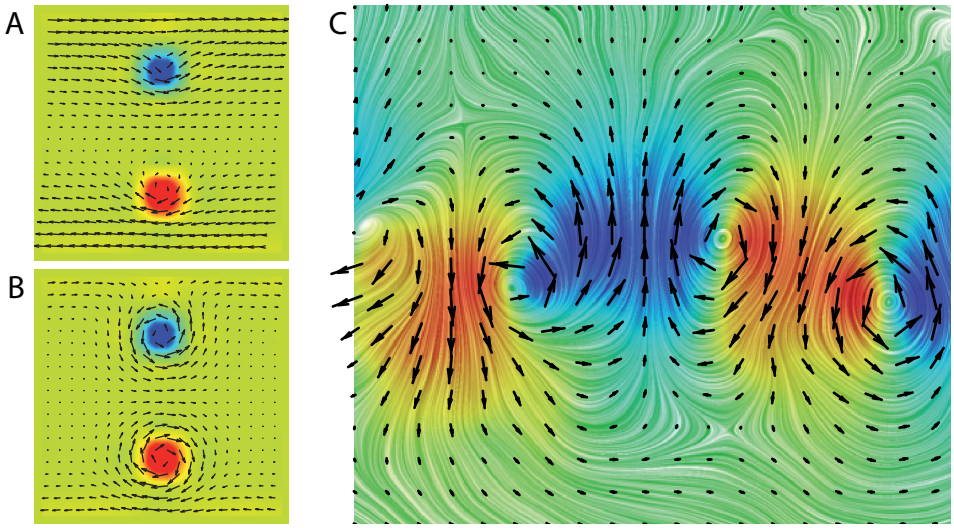

Fig. 2.29: A: Pair of counter rotating vortices in shear flow, coloured with vorticity. The vectors do not follow the circular path generally assumed for vortices. **B:** High-passed vector field. Vectors display the local relative velocity, and reveal two recirculating regions in the flow. **C:** Example for line integral convolution: Wake of an undulating flat plate (flow from right to left), LIC coloured with the v velocity component. Flow data by courtesy of René Sonntag.

circulation of a vortex, it is possible to apply the Kutta–Joukowski theorem to derive fluid dynamic forces (e. g. Henningsson & Hedenstroem, 2011). The circulation (dimension m^2/s) can be measured either by integrating the tangential velocity along a closed loop around the vortex core, or by integrating vorticity over the area of the vortex core, depending on the experimental circumstances. The choice of the integration path or the integration area is crucial (Raffel et al., 2007), and sometimes intricate. The selection of the optimal integration path can be simplified by calculating tangent velocity integrals for series of circles with increasing diameter around the centre of a vortical structure (see red circles in Figure 2.30A). When plotting the circulation over the diameter of the circles, an asymptotic convergence towards the actual circulation of the vortex can be observed (Willert & Gharib, 1991, see Figure 2.30D). The same procedure can also be performed with the area integral of vorticity to derive the actual circulation. These features are implemented in `PIVlab` and greatly simplify the determination of circulation and the application of the Kutta-Joukowski theorem.

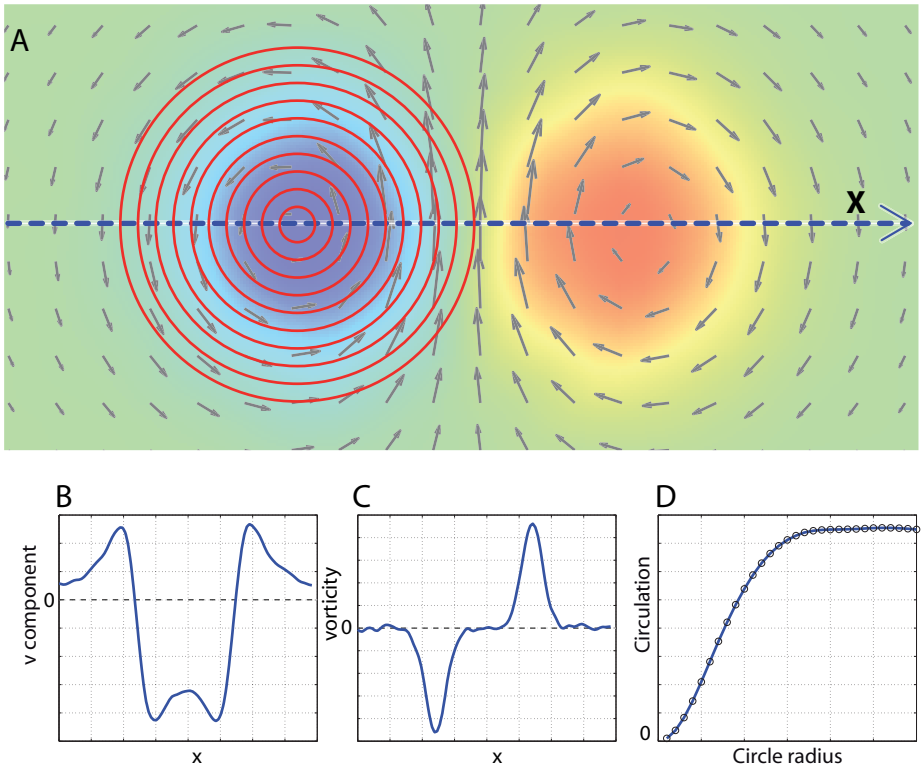

Fig. 2.30: A: Pair of counter-rotating vortices; coloured with vorticity. **B:** Normal velocity component along path x. **C:** Vorticity along path x. **D:** Circulation of one vortex plotted over circle radius.

SUMMARY

This chapter introduced the methods and principles that are required to derive accurate velocity information from two-dimensional cross-sections through a fluid. It has been shown that the choice of pre-processing, correlation and post-processing techniques influences the accuracy and the quality of these measurements. DFT with window deformation outperforms the basic DCC and DFT correlation, especially under challenging conditions. The additional computational load is compensated by the increased robustness and accuracy of the algorithm. Under optimal conditions, the bias error of the window deformation DPIV algorithms presented here is smaller than 0.005 pixels and the random error is below 0.02 pixels. When the average displacement in a DPIV study is around 6 pixels, the displacement error goes below 0.42%. Note that the final interrogation area that was selected in all the accuracy tests is only $16 \cdot 16$ pixels and using larger interrogation areas will further increase the accuracy (Raffel et al., 2007).

Several filtering techniques can be used to reject erroneous vectors and gaps in the dataset can best be interpolated using a boundary value solver. The remaining noise in the velocity map is efficiently reduced by the application of a penalized least squares smoothing technique. Finally, a selection of related data processing procedures is presented, that reduce the intricacy of complex vector maps.